

MÉTODOS DE PESQUISA DE DADOS EM MEMÓRIA PRIMÁRIA

Material da Prof. Ana Eliza

Métodos de Pesquisa em Memória Primária

Pesquisa Seqüencial

- Bases ordenadas e não-ordenadas
- Pesquisa Seqüencial Melhorada
 - Bases ordenadas
- Pesquisa Binária
 - Bases ordenadas

Pesquisa Seqüencial

- Definição

- Este método consiste em realizar a busca do elemento desejado através da comparação com cada um dos elementos da base de dados, na ordem em que estão inseridos.

Pesquisa Seqüencial

- Descrição Geral

- Compara o elemento procurado com o elemento da base de dados. Em caso de coincidência, a busca é dada por encerrada com sucesso;
- O passo anterior é repetido até que toda a base de dados seja pesquisada;
- Caso atinja o final da base de dados, conclui que a busca não obteve sucesso.

Pesquisa Seqüencial

- Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 20

20 é igual a 12? Não

20 é igual a 15? Não

20 é igual a 17? Não

20 é igual a 19? Não

20 é igual a 20? Sim

Fim da pesquisa com sucesso!

Total de passos: 05

Pesquisa Seqüencial

• Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 13

13 é igual a 12? Não

13 é igual a 15? Não

13 é igual a 17? Não

13 é igual a 19? Não

13 é igual a 20? Não

13 é igual a 21? Não

13 é igual a 27? Não

13 é igual a 37? Não

Fim da pesquisa sem sucesso!

Total de passos: 08

Pesquisa Seqüencial

• Exemplo

```
int buscaSimples (float v[ ], int n, float chave) {
    int i, achou = 0;

    for (i = 0; i <= n - 1; i++) {
        if (v[i] == chave) {
            achou = 1;
            break;
        }
    }
    if (achou == 1)
        return i; // retorna a posição
    else if (n == 0)
        return -2; // base vazia
    else
        return -1; // não achou
}
```

Pesquisa Seqüencial Melhorada

- Definição

- Este método consiste em realizar a busca do elemento desejado através da comparação com cada um dos elementos da base de dados, na ordem em que estão inseridos;
- Este método leva em consideração o fato de que os elementos da base de dados estão ordenados.

Pesquisa Seqüencial Melhorada

- Descrição Geral

- Compara o elemento procurado com o elemento da base de dados. Em caso de coincidência, a busca é dada por encerrada com sucesso; caso contrário,
- Verifica se o elemento da base de dados é maior, na ordem de classificação, que o elemento procurado. Caso seja, a busca dada por encerrada sem sucesso;
- Os passos anteriores são repetidos até que toda a base de dados seja pesquisada;
- Caso atinja o final da base de dados, conclui que a busca não obteve sucesso.

Pesquisa Seqüencial Melhorada

- Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 20

20 é igual a 12? Não => 12 é maior que 20? Não

20 é igual a 15? Não => 15 é maior que 20? Não

20 é igual a 17? Não => 17 é maior que 20? Não

20 é igual a 19? Não => 19 é maior que 20? Não

20 é igual a 20? Sim

Fim da pesquisa com sucesso!

Total de passos: 05

Pesquisa Seqüencial Melhorada

- Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 13

13 é igual a 12? Não => 12 é maior que 13? Não

13 é igual a 15? Não => 15 é maior que 13? Sim

Fim da pesquisa sem sucesso!

Total de passos: 02

Pesquisa Seqüencial Melhorada

- Exemplo

```
int buscaMelhor (float v[ ], int n, float chave) {
    int i, achou = 0;

    for (i = 0; i <= n - 1; i++) {
        if (v[i] == chave) {
            achou = 1;
            break;
        }
        else if (v[i] > chave)
            break;
    }
    if (achou == 1)
        return i; // retorna a posição
    else if (n == 0)
        return -2; // base vazia
    else
        return -1; // não achou
}
```

Pesquisa Binária

- Definição

- Este método é uma alternativa mais eficiente com relação à pesquisa seqüencial;
- Exige que os elementos da base de dados estejam ordenados.

Pesquisa Binária

- Descrição Geral

1. Compara o elemento procurado com aquele situado no meio da base de dados;
2. Se o elemento do meio é igual ao elemento procurado, então a busca é encerrada com sucesso;
3. Caso contrário, se o elemento do meio é maior do que o elemento procurado, então podemos garantir que o elemento procurado não encontra-se na segunda metade da tabela;
4. Se o elemento do meio é menor do que o elemento procurado, então podemos garantir que o elemento procurado não encontra-se na primeira metade da tabela;

Pesquisa Binária

- Descrição Geral (cont.)

5. A cada passo, a pesquisa binária reduz pela metade o número de elementos a serem considerados na busca;
6. Repetimos os passos de 1 a 4 até que o elemento seja encontrado ou até que o número de elementos a serem considerados zere.

Pesquisa Binária

- Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 20

Passo 1 => Início: 0 Fim: 7 Meio: 3

20 é igual a 19? Não => 20 é menor que 19? Não

Passo 2 => Início: 4 Fim: 7 Meio: 5

20 é igual a 21? Não => 20 é menor que 21? Sim

Passo 3 => Início: 4 Fim: 4 Meio: 4

20 é igual a 20? Sim

Fim da pesquisa com sucesso!

Total de passos: 03

Pesquisa Binária

- Exemplo

Base de Dados (12 15 17 19 20 21 27 37)

Valor procurado: 13

Passo 1 => Início: 0 Fim: 7 Meio: 3

13 é igual a 19? Não => 13 é menor que 19? Sim

Passo 2 => Início: 0 Fim: 2 Meio: 1

13 é igual a 15? Não => 13 é menor que 15? Sim

Passo 3 => Início: 0 Fim: 0 Meio: 0

13 é igual a 12? Não => 13 é menor que 12? Não

=>>> Início: 1 Fim: 0 => **Início > Fim**

=>>> Fim da pesquisa **sem** sucesso!

Total de passos: 03

Pesquisa Binária

- Exemplo

```
int buscaBinaria (float v[ ], int n, float chave) {
    int inicio = 0, meio, fim = n - 1;

    do {
        meio = (inicio + fim) / 2;
        if (v[meio] > chave) {
            fim = meio - 1;
        }
        else
            inicio = meio + 1;
    } while (chave != v[meio] && inicio <= fim);
    if (chave == v[meio])
        return meio; // retorna a posição
    else if (n == 0) {
        return -2; // base vazia
    }
    else
        return -1; // não achou
}
```

Comparação entre Métodos

- **Busca Seqüencial Simples**
 - Utilizada em bases de dados ordenadas e não ordenadas.
- **Busca Seqüencial Melhorada**
 - Utilizada apenas em bases de dados ordenadas.
- **Busca Binária**
 - Utilizada apenas em bases de dados ordenadas.

Comparação entre Métodos

- **Busca Seqüencial Melhorada**

- No caso do valor procurado pertencer à base de dados, tem a mesma eficiência que a busca seqüencial simples;
- No caso do valor procurado não pertencer à base de dados, tem eficiência superior a da busca seqüencial simples.

- **Busca Binária**

- Em média, tem eficiência superior a da busca seqüencial melhorada, em ambos os casos, ou seja, o valor procurado pertencendo ou não a base de dados.